

AMENDMENTS IN THE SPECIFICATION

Please replace the paragraph beginning on page 1, line 4, with the following:

01
The present invention is related to the subject matter of the following commonly assigned, co-pending United States Patent Applications filed concurrently herewith: Serial No. [[____]] 09/692,350 (Docket No. AUS9-2000-0621) entitled "Efficient Process for Handover Between Subnet Managers"; Serial No. [[____]] 09/692,347 (Docket No. AUS9-2000-0622) entitled "Method and System for Scalably Selecting Unique Transaction Identifiers"; Serial No. [[____]] 09/692,349 (Docket No. AUS9-2000-0623) entitled "Method And System For Reliably Defining and Determining Timeout Values In Unreliable Datagrams"; and Serial No. [[____]] 09/692,353 (Docket No. AUS9-2000-0624) entitled "Method and System For Choosing A Queue Protection Key That is Tamper-proof From An Application". The content of the above-referenced applications is incorporated herein by reference.

Please replace the paragraph beginning on page 7, line 3, with the following:

02
The present invention provides a discovery utility that allows an OS of a distributed computer system, such as a system area network (SAN), to be notified ~~when-ever~~ whenever a new component (node or device) is added to the SAN. The invention is also applicable to discovery of previously connected devices that were in the OS database but have been removed from the network for one reason or another. The following description will be understood to refer to both types of devices although specific embodiments are described with reference to a new device. In order to appreciate the environment within which the invention is preferably practiced, a description of a SAN configured with routers, switches, and end nodes, etc. is provided below. ~~Presentation of the~~ The environment and particular functional aspects of the environment which enable the invention to be practiced are provided with reference to Figures 1-5. Section headings have been provided to distinguish the hardware and software architecture of the SAN. However, those skilled in the art understand that the descriptions of either architecture necessarily includes references to both components.

Please replace the paragraph beginning on page 16, line 20, with the following:

3
G Verbs interface 207 also provide a mechanism for retrieving completed work from completion queue 411. Completion queue 411 contains Completion Queue Elements (CQEs) 413 which [[413]] contain information about previously completed WQEs 405. Completion queues [[413]] 411 are employed to create a single point of completion notification for multiple QPs 301. CQE 413 contains sufficient information to determine the QP 301 and specific WQE 405 that completed. A completion queue context (not shown) is a block of information that contains pointers to, length, and other information needed to manage individual completion queues 411.

Please replace the paragraph beginning on page 20, line 2, with the following:

2
G Referring now to Figure 8, there is illustrated an illustrative embodiment of a process of registering an OS for notification of new devices on SAN 113. The subscription process starts at block 803 and proceeds to block 805, where a consumer (e.g., an OS) that desires to be informed of changes to SAN 113 sends a Management Datagram (MAD) packet to SA 609 requesting to be informed of changes in SAN 113. Following, in block 807, SA 609 checks to see if the consumer has privileges to the paths to which the consumer is subscribing. The consumer, for example, may not have access because the paths that are being requested are owned by a partition of SAN 113 to which the consumer does not belong. Thus, in one embodiment, an OS that has signed up for notification is notified by the SA only when the new component is added to the subnet and is placed in a partition to which the OS has access.

Please replace the paragraph beginning on page 22, line 27, with the following:

5
G Referring again to Figure 6, an example of messaging (i.e., the flow of messages) among nodes on SAN 113 during operation of the invention is illustrated. Consumer OS 611 in Host node 603 desires to be notified of changes in SAN 113, for example, addition of a component via Added Node 605, so that the consumer OS 611 may support the component and provide the component with access to SAN resources, accordingly. These changes may also include removal[[s]] of components or nodes, but the operating scenario described herein focuses on the addition of components.

Please replace the paragraph beginning on page 24, line 4, with the following:

6
Accordingly, with the present invention, new nodes are dynamically discovered across SAN 113 and configured for use by the [[a]] remote consumer (or OS) that is ultimately going to support and/or control the new node. A system administrator can thus connect a new node to any location of SAN 113, and the new node is automatically provided necessary software support.
